最佳跨架構AI應用程式平台

Joel Lin, Intel
Dec. 28th, 2021
Choose the Best Accelerated Technology

Intel® oneAPI Tools for AI

Joel Lin
Dec. 28th, 2021
**Intel’s oneAPI Ecosystem**

**Built on Intel’s Rich Heritage of CPU Tools Expanded to XPU**s

**oneAPI**
A cross-architecture language based on C++ and SYCL standards

Powerful libraries designed for acceleration of domain-specific functions

A complete set of advanced compilers, libraries, and porting, analysis and debugger tools

**Powered by oneAPI**
Frameworks and middleware that are built using one or more of the oneAPI industry specification elements, the DPC++ language, and libraries listed on oneapi.com.

Visit [software.intel.com/oneapi](http://software.intel.com/oneapi) for more details. Some capabilities may differ per architecture and custom-tuning will still be required. Other accelerators to be supported in the future.
Intel oneAPI Software Tools for AI & Analytics

Intel® oneAPI Toolkits 2022 is released this December. Find out more details on the online release notes.

**Intel® oneAPI Toolkits**

**Intel® oneAPI AI Analytics Toolkit**
Accelerate machine learning & data science pipelines with optimized deep learning frameworks & high-performing Python libraries
- Data Scientists, AI Researchers, DL/ML Developers

**Intel® oneAPI Base Toolkit**
Incl. Intel® oneAPI Deep Neural Network Library (oneDNN), Intel® oneAPI Collective Communications Library (oneCCL), & Intel® oneAPI Data Analytics Library (oneDAL)
- Optimize primitives for algorithms and framework development
- DL Framework Developers - Optimize algorithms for Machine Learning & Analytics

**Toolkit Powered by oneAPI**

**Intel® Distribution of OpenVINO™ Toolkit**
Deploy high performance inference & applications from edge to cloud
- AI Application, Media, & Vision Developers

Intel® oneAPI Toolkit 2022 is released this December. Find out more details on the online release notes.
Journey to Production AI

1. Frame Opportunity
2. Hypotheses
3. Data Preparation
4. Experiment with topologies
5. Tune hyper-parameters
6. Support inference inputs
7. Document results

Development Cycle:
- Labor-intensive
- Compute-intensive (Model Training)
- Labor-intensive

Deployment Cycle:
- Optimize
- Test
- Package
- Integrate
- Deploy

Data Exploration
Train & Infer

Inference within broader application

Iteration

Evaluation

TOTAL TIME TO SOLUTION (TTS)

Process validated with customer in Oil & Gas sector
AI Software Stack for Intel XPU

Intel offers a Robust Software Stack to Maximize Performance of Diverse Workloads

---

**E2E Workloads**
- Census
- NYTaxi
- Mortgage

**Intel® Low Precision Optimization Tool**

**Model Zoo for Intel® Architecture**

**Open Model Zoo**

---

**DL/ML Tools**

**DL/ML Middleware & Frameworks**

---

**DL/ML Tools**
- TensorFlow
- PyTorch
- OpenVINO

**DL/ML Middleware & Frameworks**
- Model Optimizer
- Inference Engine

---

**Libraries & Compiler**
- DPC++ / DPPY
- oneMKL
- oneDAL
- oneTBB
- oneCCL
- oneDNN
- oneVPL

---

Part of the Intel® oneAPI Base Toolkit
AI Software Stack for Intel XPU

Intel offers a Robust Software Stack to Maximize Performance of Diverse Workloads

- Intel® oneAPI AI Analytics Toolkit
  - Develop DL models in Frameworks, ML & Analytics in Python
  - numba, pandas, Modin, scipy, daal4Py, xgboost, TensorFlow, PyTorch

- Intel® oneAPI Base Toolkit
  - Kernel Selection, Write, Customize Kernels
  - DPC++, oneMKL, oneDNN, oneVPL

- Intel® Low Precision Optimization Tool
  - Intel® openVINO™ Toolkit
  - Model Zoo for Intel® Architecture
  - Open Model Middleware & Frameworks
  - Model Taxi, Census, Mortgage...

- Full Set of Intel oneAPI cross-architecture AI ML & DL Software Solutions
Intel® oneAPI Base Toolkit

Accelerate Data-centric Workloads

A set of core tools and libraries for developing high-performance applications on Intel® CPUs, GPUs, and FPGAs

Intel® oneAPI IoT Toolkit 2022 is out. Find out more details on the release notes.

Who Uses It?
- A broad range of developers across industries
- Native Code Developers/Framework Developers

Top Features/Benefits
- Data Parallel C++ (DPC++) compiler, library and analysis tools; DPC++ Compatibility tool helps migrate existing code written in CUDA
- Optimized performance libraries for threading, math, data analytics, deep learning, and video/image/signal processing

Learn More: intel.com/oneAPI-BaseKit
AI Development Cycle
Intel® oneAPI AI Analytics Toolkit

Accelerates end-to-end Machine Learning and Data Analytics pipelines with frameworks and libraries optimized for Intel® architectures

Target Model building, evaluating, training and datap preprocessing and data analytics in AI Development Cycle

Who Uses It?

Data scientists, AI Researchers, Machine and Deep Learning developers, AI application developers

Learn More: intel.com/oneAPI-AIKit
Performance Benefits

Maximize Hardware Value with Intel-optimized Software
Deep Learning Training & Inference Performance
Uses Intel® Optimization for PyTorch with 3rd Gen Intel® Xeon® Scalable Processors

<table>
<thead>
<tr>
<th>Training</th>
<th># Cores per instance</th>
<th># Instances</th>
<th>BF16 (samples/s)</th>
<th>FP32 (samples/s)</th>
<th>Speedup Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>DLRM</td>
<td>28</td>
<td>1</td>
<td>99321</td>
<td>71061</td>
<td>1.40</td>
</tr>
<tr>
<td>ResNet-50</td>
<td>28</td>
<td>4</td>
<td>399</td>
<td>243</td>
<td>1.64</td>
</tr>
<tr>
<td>ResNeXt-101</td>
<td>28</td>
<td>4</td>
<td>193</td>
<td>120</td>
<td>1.60</td>
</tr>
<tr>
<td>32x4d</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1. BF16 training performance gains over baseline (FP32 with Intel oneDNN)

<table>
<thead>
<tr>
<th>Inference</th>
<th># Cores per instance</th>
<th># Instances</th>
<th>INT8 (samples/s)</th>
<th>FP32 (samples/s)</th>
<th>Speedup Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>DLRM</td>
<td>1</td>
<td>28</td>
<td>611082</td>
<td>214559</td>
<td>2.85</td>
</tr>
</tbody>
</table>

Table 2. INT8 inference performance gains over baseline (FP32 with Intel oneDNN)

ML Performance with Intel-optimized scikit-learn

```python
from sklearn.svm import SVC
X, Y = get_dataset()
clf = SVC().fit(X, y)
res = clf.predict(X)
```

Common Scikit-learn (mainline)

```python
import daal4py as d4p
d4p.patch_sklearn()
from sklearn.svm import SVC
X, Y = get_dataset()
clf = SVC().fit(X, y)
res = clf.predict(X)
```

Scikit-learn on Intel CPU optimized by Intel® oneAPI AI Analytics Toolkit

Same Code, Same Behavior

*Scikit-learn, not scikit-learn-like*
End-to-End Data Pipeline Acceleration

- **Workload:** Train a model using 50 years of Census dataset from IPUMS.org to predict income based on education

- **Solution:** Intel Modin for data ingestion and ETL, Daal4Py and Intel scikit-learn for model training and prediction

- **Performance Gains**
  - Read_CSV (Read from disk and store as a dataframe): 6x
  - ETL operations: 38x
  - Train Test Split: 4x
  - ML training (fit & predict) with Ridge Regression: 21x

AI Deployment Cycle
High-Performance Deep Learning Using Intel® Distribution of OpenVINO™ toolkit - Powered by oneAPI

A toolkit for fast, more accurate real-world results using high-performance AI and computer vision inference deployed into production on Intel XPU architectures (CPU, GPU, FPGA, VPU) from edge to cloud

Who needs this product?
AI application developers, OEMs, ISVs, System Integrators, Vision and Media developers

Top Features/Benefits
High-performance, deep learning inference deployment
Streamlined development; ease of use
Write once, deploy anywhere

Proven, industry-leading accelerated technology

software.intel.com/openvino-toolkit
Which Toolkit Should I Use
Use Both!
Intel® oneAPI Analytics Toolkit & Intel® Distribution of OpenVINO™ toolkit

Toolkits are complementary to each other and recommendation is to use them both based on your current phase of AI Journey

- I am exploring and analyzing data; I am developing models
- I want performance and compatibility with frameworks and libraries I use
- I would like to have drop-in acceleration with little to no additional code changes
- I prefer not to learn any new tools or languages

- I am deploying models
- I want leading performance and efficiency across multiple target HW
- I'm concerned about having lower memory footprint, which is critical for deployment
- I am comfortable with learning and adopting a new tool or API to do so

Data Scientist/ML Developer
Intel® oneAPI AI Analytics Toolkit

App Developer
Intel® Distribution of OpenVINO™ toolkit

If you prefer working on primitives and to optimize kernels and algorithms directly using oneAPI libraries (oneDNN, oneCCL & oneDAL), then use Intel® oneAPI Base Toolkit
Accrad AI-based Solution Helps Accelerate Lung Disease Diagnosis
Optimized by Intel® oneAPI Analytics Toolkit & Intel® Distribution of OpenVINO™ toolkit

Al Machine Vision Disease Detection

*CheXRad* is a machine learning edge application that helps radiologists and physicians identify COVID-19, viral pneumonia and other diseases on chest X-ray images, and predict the need for ventilators.

- *CheXRad* comes pre-configured with a COVID-19 and viral pneumonia classification neural network.
- To architect, train and validate the neural network, Accrad used Intel Tensorflow from AI Analytics Toolkit and the infrastructure provided by Intel oneAPI DevCloud to develop the model.
- To optimize its model for deployment, Accrad used OpenVINO™ toolkit and Intel® DevCloud for Edge.
- *CheXRad* could label pathologies in 140 chest x-rays in just 90 seconds — up to 160x faster than radiologists, at comparable levels of accuracy, sensitivity and specificity.

“*With the help of Intel, we were able to train, optimize, and deploy a machine learning model in less time and at a lower operational cost than available alternatives, enabling us to get to market fast with a powerful solution that’s optimized for Intel® architecture.*” – Moloti Nakampe, R&D Director

Learn more in this [solution brief](#)
AbbVie Machine Translation Solution
Optimized by Intel® oneAPI Analytics Toolkit & Intel® Distribution of OpenVINO™ toolkit

A research-based biopharmaceutical company powered by Intel® Xeon® processors.

**Abbelfish Machine Translation** uses Intel® Optimization for TensorFlow of AI Analytics Toolkit
- A custom model that provides more accurate translations than commercially available ones. Model includes 24 layers and 500+ million parameters that took more than 4 months to train
- Intel TensorFlow provided a greater performance boost while AbbVie did not have to change its code or APIs from standard TensorFlow

**AbbVie Search** uses Intel® Distribution of OpenVINO™ toolkit
- OpenVINO toolkit provided great speed-up to answer questions from a scientific article or clinical report when compared to standard TensorFlow
- Requires scaling across the company, so uses OpenVINO™ Model Server to serve inferences

AI Containers for Flexibility

- Optimized, validated, deployable AI containers and artifacts for Intel® platforms
- Available via Docker containers and Intel software stacks. Will expand to include Kubernetes orchestrations, Helm charts, AI models, pipelines and more
- Access oneContainer Portal

Key Models (GPU-WIP, Check the portal for latest availability)

<table>
<thead>
<tr>
<th>Topology</th>
<th>Framework</th>
<th>Topology</th>
<th>Framework</th>
</tr>
</thead>
<tbody>
<tr>
<td>DLRM</td>
<td>PYT</td>
<td>Mask R-CNN</td>
<td>PYT, TF, OV</td>
</tr>
<tr>
<td>ResNet50</td>
<td>PYT, TF, OV</td>
<td>RNN-T</td>
<td>PYT, TF, OV</td>
</tr>
<tr>
<td>BERT-large</td>
<td>PYT, TF, OV</td>
<td>3D-UNet</td>
<td>TF, OV</td>
</tr>
<tr>
<td>Transformer-LT</td>
<td>PYT, TF</td>
<td>DIEN</td>
<td>TF</td>
</tr>
<tr>
<td>MobileNet-v1</td>
<td>PYT, TF, OV</td>
<td>Wide &amp; Deep</td>
<td>PYT, TF</td>
</tr>
<tr>
<td>SSD-MobileNet-v1</td>
<td>PYT, TF, OV</td>
<td>RNX101</td>
<td></td>
</tr>
<tr>
<td>SSD-Resnet34</td>
<td>PYT, TF, OV</td>
<td>Yolo-V3</td>
<td>PYT, TF, OV</td>
</tr>
<tr>
<td>WaveNet*</td>
<td>TF</td>
<td>NCF*</td>
<td>TF</td>
</tr>
</tbody>
</table>
Key Takeaways & Call to Action

- Intel toolkits are FREE, complementary & work seamlessly together
- They help achieve performance & efficiency across different stages of AI Journey
- Recommend the toolkits based on current phase of customer pipeline

Download the toolkits

Intel® oneAPI AI Analytics Toolkit
Intel® Distribution of OpenVINO ™ toolkit
Intel® oneAPI Base Toolkit

Learn more about Intel® oneAPI Toolkits
intel.com/oneAPI-AllToolkits
BackUp
AI Development Workflow

While there are a few distribution options to directly download Intel-optimized FWKs, machine learning libraries and tools individually. Our recommendation is to get them via the Intel® oneAPI AI Analytics Toolkit for seamless interoperability and good out-of-box experience.

Determine Use Case

Data Analytics
- Data Ingestion & Pre-processing
  - Use AI Kit (Modin, Omnisci, Pandas, Numpy, Scipy)

Machine Learning
- Classical ML Training & Prediction
  - Use AI Kit (Scikit-learn+Daal4py, XGBoost)
- Optimize Primitives for DL FWKs
  - Use Base Kit (oneDNN, oneCCl)

Deep Learning
- Train DL Model on Intel CPU/dGPU
  - Use AI Kit (Intel-optimized TensorFlow, Pytorch)
- Re-train a model on custom data
  - Use AI Kit (Intel-optimized TensorFlow, Pytorch)
- Pick a Pre-trained Intel-optimized Model
  - Use AI Kit (Model Zoo for Intel® architecture)
- Further optimize
  - Convert to Low Precision & run inference
  - Run DL Inference on trained model
  - Use AI Kit (Model Zoo for Intel® architecture)

Pick a pre-trained model in IR format (Open Model Zoo)

Trained Model

Deploy DL Models on Intel® platforms
- Use OpenVINO™ Toolkit (Model Optimizer, Inference Engine, Deployment Manager, etc.)

Public models trained with any FWK – TF, Caffe, ONNX, MXNet, etc.

AI Kit = Intel® oneAPI AI Analytics Toolkit
Base Kit = Intel® oneAPI Base Toolkit

Native Code Developers, Framework Developers
Data Scientists, AI Researchers, ML/DL Developers
AI, Media & Computer Vision Application Developers
Getting Started with Intel® oneAPI AI Analytics Toolkit

Overview
- Visit Intel® oneAPI AI Analytics Toolkit (AI Kit) for more details and up-to-date product information
- Release Notes

Installation
- Download the AI Kit from Intel, Anaconda or any of your favorite package managers
- Get started quickly with the AI Kit Docker Container
- Installation Guide
- Utilize the Getting Started Guide

Hands on
- Code Samples
- Build, test and remotely run workloads on the Intel® DevCloud for free. No software downloads. No configuration steps. No installations.

Learning
- Machine Learning & Analytics Blogs
- Intel AI Blog site
- Webinars & articles

Support
- Ask questions and share information with others through the Community Forum
- Discuss with experts at AI Frameworks Forum

Download Now
Getting Started with Intel® Distribution of OpenVINO™ Toolkit

**Overview**
- Visit Intel® Distribution of OpenVINO Toolkit for more details and up-to-date product information
- Release Notes

**Installation**
- Download the Intel® Distribution of OpenVINO™ toolkit, or get via YUM or APT repositories
- Utilize the Getting Started Guide

**Hands on**
- Understand sample demos and tools included
- Build, test and remotely run workloads on the Intel® DevCloud for the Edge before buying hardware

**Learning**
- Intel AI Blog site
- Webinars & articles
- Choose hardware option with Performance Benchmarks

**Support**
- Ask questions and share information with others through the Community Forum

*Download Now*
Performance varies by use, configuration and other factors. Learn more at www.Intel.com/PerformanceIndex.

Performance results are based on testing as of dates shown in configurations and may not reflect all publicly available updates. See backup for configuration details. No product or component can be absolutely secure.

Your costs and results may vary.

Intel technologies may require enabled hardware, software or service activation.

Intel does not control or audit third-party data. You should consult other sources to evaluate accuracy.

© Intel Corporation. Intel, the Intel logo, and other Intel marks are trademarks of Intel Corporation or its subsidiaries. Other names and brands may be claimed as the property of others.
Deep Learning Training and Inference Performance using Intel® Optimization for PyTorch with 3rd Gen Intel® Xeon® Scalable Processors


**ResNet50/ResNext101 dataset (FP32/BF16):** ImageNet Dataset

**DLRM batch size (FP32/BF16):** 2K-instance, 1 instance

**DLRM dataset (FP32/BF16):** Criteo Terabyte Dataset

**DLRM batch size (INT8):** 16-instance, 28 instances, dummy data.

Tested by Intel as of 6/2/2020.

**Intel® Xeon® Platinum 8380H Processor,** 4 socket, 28 cores HT On Turbo ON Total Memory 768 GB (24 slots/ 32GB/ 3200 MHz), BIOS: WLYDCRB1.SYS.0015.P96.2005070242 (unicode: 0x700001b), Ubuntu 20.04 LTS, kernel 5.4.0-29-generic

**PyTorch:** https://github.com/pytorch/pytorch.git

**Intel Extension for PyTorch:** https://github.com/intel/intel-extension-for-pytorch.git

**gcc:** 8.4.0,

**Intel® oneAPI Deep Neural Network Library (oneDNN) version:** v1.4


**ResNext101 32x4d:** https://github.com/intel/optimized-models/tree/master/pytorch/ResNet101_32x4d

**DLRM:** https://github.com/intel/optimized-models/tree/master/pytorch/dlrm

**Inference Throughput FP32 vs Int8 optimized by Intel® Optimization for Tensorflow and Intel® Low Precision Optimization Tool** (part of the Intel® oneAPI AI Analytics Toolkit)

Tested by Intel as of: 10/26/2020: Tensorflow v2.2 (https://github.com/intel-tensorflow/tensorflow/tree/v2.2.0); Compiler: GCC 7.2.1; DNNL(https://github.com/oneapi-src/oneDNN) v1.2.0 75d0b1a7f3586c212e37aceebb9ad221ceee7216; Dataset: ImageNet/Coco/Dummy, refer to each model README; Precision: FP32 and Int8

**Platform:** Intel® Xeon® Platinum 8280 CPU; #Nodes: 1; #Sockets: 2; Cores/socket: 28; Threads/socket: 56; HT: On; Turbo: On; BIOS version: SES5620.86B.02.01.0010.010620200716; System DDR Mem Config: 12 slots / 16GB / 2933; OS: CentOS Linux 7.8; Kernel: 4.4.240-1.el7.elrepo.x86_64

**Stock scikit-learn vs Intel-optimized scikit-learn**

Testing by Intel as of 10/23/2020. Intel® oneAPI Data Analytics Library 2021.1 (oneDAL), scikit-learn 0.23.1, Intel® Distribution for Python 3.8; Intel® Xeon® Platinum 8280LCPU @ 2.70GHz, 2Sockets, 28 cores per socket, 10M samples, 10 features, 100 clusters, 100 iterations, float32

Intel does not control or audit third-party data. You should consult other sources to evaluate accuracy.
XGBoost CPU vs GPU
Test configs: Tested by Intel as of 10/13/2020;
CPU: c5.18xlarge AWS Instance (2 x Intel® Xeon Platinum 8124M @ 18 cores, OS: Ubuntu 20.04.2 LTS, 193 GB RAM. GPU: p3.2xlarge AWS Instance (GPU: NVIDIA Tesla V100 16GB, 8 vCPUs), OS: Ubuntu 18.04.2 LTS, 61 GB RAM. SW: XGBoost 1.1:build from sources. compiler – G++ 7.4, nvcc 9.1. Intel® Data Analytics Acceleration Library (Intel® DAAL): 2019.4 version; Python env: Python 3.6, Numpy 1.16.4, Pandas 0.25, Scikit-learn 0.21.2.

XGBoost fit CPU acceleration
Test configs: Tested by Intel as of 10/13/2020; c5.24xlarge AWS Instance, CLX 8275 @ 3.0GHz, 2 sockets, 24 cores per socket, HT: on, DRAM (12 slots / 32GB / 2933 MHz); SW: XGBoost 0.81, 0.9, 1.0 and 1.1: build from sources. compiler – G++ 7.4, nvcc 9.1. Intel® DAAL: 2019.4 version; Python env: Python 3.6, Numpy 1.16.4, Pandas 0.25, Scikit-learn 0.21.2.

End-to-End Census Workload Performance
Tested by Intel as of 10/15/2020. 2x Intel® Xeon® Platinum 8280 @ 28 cores, OS: Ubuntu 19.10.5.3.0-64-generic Mitigated, 384GB RAM. SW: Modin 0.8.1, scikit-learn 0.22.2, Pandas 1.0.1, Python 3.8.5, Daal4Py 2020.2 Census Data, (21721922, 45). Dataset is from IPUMS USA, University of Minnesota, www.ipums.org. Version 10.0.

Tiger Lake + Intel® Distribution of OpenVINO™ toolkit vs Coffee Lake CPU

<table>
<thead>
<tr>
<th>System Board</th>
<th>Intel prototype, TGL U DDR4 SODIMM RVP</th>
<th>ASUSTeK COMPUTER INC. / PRIME Z370-A</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>11th Gen Intel® Core™ i5-1145G7E @ 2.6 GHz.</td>
<td>8th Gen Intel® Core™ i5-8500T @ 3.0 GHz.</td>
</tr>
<tr>
<td>Sockets / Physical cores</td>
<td>1 / 4</td>
<td>1 / 6</td>
</tr>
<tr>
<td>HyperThreading / Turbo Setting</td>
<td>Enabled / On</td>
<td>Na / On</td>
</tr>
<tr>
<td>Memory</td>
<td>2 x 8198 MB 3200 MT/s DDR4</td>
<td>2 x 16384 MB 2667 MT/s DDR4</td>
</tr>
<tr>
<td>OS</td>
<td>Ubuntu* 18.04 LTS</td>
<td>Ubuntu* 18.04 LTS</td>
</tr>
<tr>
<td>Kernel</td>
<td>5.8.0-050800-generic</td>
<td>5.3.0-24-generic</td>
</tr>
<tr>
<td>Software</td>
<td>Intel® Distribution of OpenVINO™ toolkit 2021.1.075</td>
<td>Intel® Distribution of OpenVINO™ toolkit 2021.1.075</td>
</tr>
<tr>
<td>BIOS</td>
<td>Intel TGLIFU11.R00.3243.A04.2006302148</td>
<td>AMI, version 2401</td>
</tr>
<tr>
<td>BIOS release date</td>
<td>Release Date: 06/30/2020</td>
<td>7/12/2019</td>
</tr>
<tr>
<td>BIOS Setting</td>
<td>Load default settings</td>
<td>Load default settings, set XMP to 2667</td>
</tr>
<tr>
<td>Test Date</td>
<td>9/9/2020</td>
<td>9/9/2020</td>
</tr>
<tr>
<td>Precision and Batch Size</td>
<td>CPU: INT8, GPU: FP16-INT8, batch size: 1</td>
<td>CPU: INT8, GPU: FP16-INT8, batch size: 1</td>
</tr>
<tr>
<td>Number of Inference Requests</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>Number of Execution Streams</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>Power (TDP Link)</td>
<td>28 W</td>
<td>35W</td>
</tr>
</tbody>
</table>
Thank You.